

CLAIMS

What is claimed is:

1. A method for reducing contention in a multithreaded processor; the method comprising the step of
 - (a) providing at least one thread stack/thread heap combination in an address space on the processor, wherein the thread heap is for thread local memory usage and wherein the thread stack and thread heap grow in opposite directions.
2. The method of claim 1 in which the at least one thread stack/thread heap combination comprises at least two thread stack/thread heap combinations.
3. The method of claim 2 which includes the step of (b) providing a dead zone between the at least two thread stack/thread heap combination, wherein data can not be written to, read from and executed from the dead zone.
4. The method of claim 1 wherein the dead zone providing step (a) further comprises the steps of:
 - (a1) providing a base address for the thread stack
 - (a2) creating an initialization for the thread heap from the base address; and
 - (a3) assigning memory regions in the address space to the thread stack and thread heap.

5. The method of claim 4 which includes the step of (a4) allowing the assigned memory regions of the thread stack and thread heap to grow in opposite directions as needed.

6. A computer readable medium containing program instructions for reducing contention in a multithreaded processor; the program instructions for:

(a) providing at least one thread stack/thread heap combination in an address space on the processor, wherein the thread heap is for thread local memory usage and wherein the thread stack and thread heap grow in opposite directions.

7. The computer readable medium of claim 6 in which the at least one thread stack/thread heap combination comprises at least two thread stack/thread heap combinations.

8. The computer readable medium of claim 7 which includes program instructions for (b) providing a dead zone between the thread stack/thread heap combination, wherein data can not be written to, read from and executed from the dead zone.

9. The computer readable medium of claim 6 wherein the dead zone providing program instruction (a); further comprise the steps of:

(a1) providing a base address for the thread stack
(a2) creating an initialization for the thread heap from the base address; and
(a3) assigning memory regions in the address space to the thread stack and thread heap.

10. The computer readable medium of claim 9 which includes program instructions for (a4) allowing the assigned memory regions of the thread stack and thread heap to grow in opposite directions as needed.

11. A system for reducing contention in a multithreaded processor; the system comprising:

means for providing at least one thread stack/thread heap combination in an address space on the processor, wherein the thread heap is for thread local memory usage and wherein the thread stack and thread heap grow in opposite directions.

12. The system of claim 11 in which the at least one thread stack/thread heap combination comprises at least two thread stack/thread heap combinations.

13. The system of claim 12 which includes means for providing a dead zone between the thread stack/thread heap combination, wherein data can not be written to, read from and executed from the dead zone.

14. The system of claim 11 wherein the dead zone providing means further comprises:

means for providing a base address for the thread stack

means for creating an initialization for the thread heap from the base address;

and means for assigning memory regions in the address space to the thread stack and thread heap.

15. The method of claim 14 which includes means for allowing the assigned memory regions of the thread stack and thread heap to grow in opposite directions as needed.